

Morpho - Bug #555

extend windowing framework to allow multiple windows with menus

07/19/2002 11:22 AM - Matt Jones

Status:	Resolved	Start date:	07/19/2002
Priority:	Immediate	Due date:	
Assignee:	Matt Jones	% Done:	0%
Category:	morpho - general	Estimated time:	0.00 hour
Target version:	1.2	Spent time:	0.00 hour
Bugzilla-Id:	555		
Description			
<p>The existing ClientFramework is designed to have a single application window for morpho that contains the menu and toolbars. In the redesign of the UI to make it more data centric, we have decided we need for each package and resultset to be opened in its own window and to include a common set of menu bars and toolbars and status bars.</p> <p>This bug is to redesign the ClientFramework architecture so that it allows for plugins to create new windows with a specific content pane registered, and the framework would return the window that was created, as well as notify existing windows to update their window menus with the presence of the new window.</p> <p>This new "createWindow" function would be called by the data package and query subsystems to create their windows, and possibly by the editor subsystem to create its windows as well.</p> <p>The redesign will likely involve creating a new "Morpho" class that represents the application as a whole and includes utility and main methods and application state fields, and a reworked ClientFramework that implements the visual Frame components separately.</p>			

History

#1 - 08/19/2002 05:11 PM - Matt Jones

I have largely finished merging the MULTIDOC_BRANCH onto the head. This means that the main application class is now edu.ucsb.nceas.morpho.Morpho, and ClientFramework is gone. I have a few leftover bugs to fix (e.g., some images aren't loading, and morpho doesn't think Metacat is available because the "doPing" method was commented out for expediency, etc.). I'll clear these off the list tomorrow, but I wanted you to be able to start back into your modules on the head.

The query subsystem is largely working -- Jing has a list of minor todo's. Dan's DATACENTRIC_DATAPACKAGE branch will need to be merged. Given that the old implementation doesn't separate the UI much (GUI classes extend JFrame and do a bunch of other stuff too), there will probably be some pain involved in this conversion. The editor plugin has been converted, but does not make use of the new windowing system (see below) because the DocFrame class was too much for me to understand easily. I have to convert DocFrame so it no longer extends JFrame, and this is going to take a while given that it does a ton of non-ui stuff in there.

Below you'll find a brief overview of the new interfaces. See the more detailed documentation in the MorphoFramework.svg and morpho-design.vsd design documents if you want to know more.

The user interface parts of ClientFramework have been transformed into classes that support multiple windows with menus, toolbars, and the status bar in each. You can get a new window for use in your application by using a call like this:

```
MorphoFrame window = UIController.getInstance().addWindow("Some random title");
```

and you can close it using:

```
window.close();
```

These methods take care of updating the menus and other application wide housekeeping, so you should use them religiously (ie, never instantiate a MorphoFrame on your own).

The Services infrastructure has also changed. You now register a service using:

```
ServiceController.getInstance().addService(SomeInterface.class, this);
```

and you look up a service using:

```
ServiceController.getInstance().getService(SomeInterface.class);
```

You'll have to catch some exceptions in there, but that's basically it. Note that both UIController and ServiceController are singletons so it is always safe to call their getInstance() methods in a static environment.

I've also factored out the debugging/logging code to a new class called "edu.ucsb.nceas.morpho.util.Log" that you can call statically like this:

```
Log.debug(9, "Whatever I need to output");
```

There's a bunch of other factoring that could and should be done on the Morpho class (its still a mishmash), but this was the minimum that I could do to factor out the JFrame parts. Hopefully we can come back to it and factor out the networking code into a distinct network package sometime. But we'll see. Enjoy, and comments appreciated...

#2 - 08/22/2002 11:49 AM - Matt Jones

This is now complete. Some side issues have cropped up, but they have been entered as separate bugs (see bug 569, bug 570, bug 571). DONE.

#3 - 03/27/2013 02:14 PM - Redmine Admin

Original Bugzilla ID was 555